Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

OSI SYM KIM AIM ATARI APPLE UK101 ORANGE

Registered by Australia Post Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol. 2 No.7 April 1982

A reminder to all SYM and other 6502 users. Andrew Stephanou has arranged for Prof. 'LUX' Luxenberg to change his itinery and attend the April meeting. Prof. LUX is the editor of SYM-PHYSIS and will have a lot of information to interest 6502 users.

Our beloved leader has returned and will be chairing the April meeting. We may be able to persuade him, with some difficulty, to give his opinion of America in general and Texas in particular. David and the hardware committee are looking for their next project and this will be discussed at the April meeting.

AND NOW FOR THE BIG NEWS!!! David Tasker's video board will be available from George early in May. The board will be available to KAOS members at:-

1. Board only $45.00
2. Partial Kit includes all IC's to construct the board except those which can be used from existing video circuitry. $130.00
3. Complete Kit $145.00

Connector cable from S/B to video board $16.00
Dabug upgrade to 64X32 format $10.00

Incidently, the colour photographs of the video board which you received with this newsletter are courtesy of David Tasker. He had a roll of light damaged paper which he could not use in his film processing business, so he ran of five hundred copies for us to distribute.

The next meeting will be held on Sunday 25th April at 2pm at the Essendon Primary School, corner of Raleigh St and Nicholson St, Essendon. Would the usual early arrivers please note that the children from the school will be in early for their lesson.

Remember that there will be a printer available at the meeting for anyone who wants a listing of their program. Bring along your program on 300 baud cassette or 5¼" disk. Also, for those of you who are too polite to yell out and get attention at question time, there will be a pad and pencil available, write out your question and leave it on Rosemary's table and it will be answered at the meeting or in the newsletter.

FIRST IMPRESSIONS OF THE VIC 20

I took delivery of my new VIC 20 the other day and I can tell you, it was with great anticipation that I unpacked it and plugged it in. After all the publicity and advertising hype, it had better be good. Well, it wasn't long before I discovered what all the fuss was about. I had just loaded the program and was about to test my reflexes on a game of "CAR CHASE" when WHAMO, I was shoved to the back of the crowd which had gathered while my back was turned. Since then, the only time I get a go on the thing is when the kids are in the bath and the wife's on the phone. No, that's is not fair, I'm allowed to use it any time I like between the hours of 11pm and 7am. Perhaps this is what they mean by "home" computer.

First impressions of the machine are that it is compact, strong. colourful, noisy, and dead simple to set up and operate. My VIC is stock standard. It came with a 9 volt power pack, an RF modulator, a very well written hand book which is designed for first time users and 3.5K of usable RAM. The only optional extra I purchased was the Commadore digital recorder (at $99.00 thank you very much). I guess many people will find it handy to be able to plug it into the lounge room telly using the modulator. This results in reasonable quality picture and sound, but as I have a portable colour TV which has been converted to a monitor, I chucked away the regulator, made up a lead, and went for direct audio and video. The improvement in quality was quite noticable.

The low resolution graphics (22X22), colour, and sound are quite flexible and easy to use. The eight border and foreground colours, and 64 graphics characters are right there on the keyboard, while the sixteen background colours 3 tone generators, each with 4 octave range, and a morse channel can be easily accessed with BASIC, PEEK's and POKE's. High resolution graphics (176 X184) is also possible by cunning use of the user definable character set, but on the standard VIC you are only left with 1K and the use of half the screen and the BASIC routines to drive it are rather clumsy. For $50.00 however, you can buy a plug in ROM-PAK called "SUPER EXPANDER" which extends BASIC to include graphics and sound commands and gives you an extra 3K of RAM and the use of the full screen for Hi-Res.

The keyboard is very conveniently laid out for BASIC programmers, requiring minimal use of the SHIFT keys if the code is kept purely standard. Second and third functions of keys are generally to do with graphics and colour. Basic key words can generally be abreviated to the first two or three letters, eg. PshiftO for POKE.

Loading and saving programs on the cassette recorder is a breeze as the VIC prompts you all the way, telling you exactly which buttons to press. The tape drive is even stopped automatically at the end of each operation. 'Off the shelf' program cartridges or 'ROM-PAKS' can be purchased ($50.00 each) and plug directly into the edge connector at the back. Simply switch on and you're in program. These cartridges "piggy back" into the expansion module if you are rich enough to afford one.

On the negative side, there is no machine language monitor on board, but I am told one will be available on cartridge soon. Even though the standard VIC appears cheap at $399.00 , to build it up with genuine Commadore bits is a very expensive business. For example, a single disk drive and controller sells at $800.00!

Fortunately, Commadore have been very obligingabout providing information about the VIC and one publication, "VIC REVEALED" by Nick Hampshire, is a gold mine for experienced and serious programmers, and according to George, gives plenty of clues on how to design your own hardware mods.

George assures me that alternatives to the high priced Commadore bits do not represent too great a problem. In fact you can already buy a plug in 24K RAM expansion module for around $240.00.

As a school teacher, I am looking at the VIC in terms of its value as an educational tool. So far, from what I have seen, its potential in this regard both in the home and in the classroom, is enormous. George thinks there is also enough potential in it for the hardware buffs and sees it as a replacement for the good old Superboard.

Noel Dollman

****

THE MISSING PAGE IN GEORGE'S DOS

Maybe George and David thought we could guess a lot of the answers that I asked myself the night I got home with COMP-DOS 1.2 It also just happened to be a night that George wasn't working back. So I had to puddle through by myself.

So as a helper to anyone who has just purchased the package, here is what I learnt.

COMMANDS

loading from cassette	DISK!"IO 01"
saving to file name	DISK!"PUT FRED"
where FRED is a file name.	
saving to a track	DISK!"PUT 19"
where 19 is the track number.	
Printing out to a printer (ACIA)	PRINT#1,

The new DOS can be added to the old 3.2 DOS as an extra programme, but in so doing the only advantage is that you get a 48 X 12 format. But after using 1.2 as sold by George, who wants 3.2.

On the other hand, I would like to see George write a small mod for the NEWDOS which would allow my sister to use the system more easily. In short, this would be to delete the opening choices and go direct to the listing the directory in a 48 X 12 format. (Maybe for the next issue, please.)

The other points I miss are firstly the use of control-shift to freeze the screen and secondly a POKE to salvage a programme in the memory that hasn't been put on disk.

In the last analysis, it's a great addition to OS65D3 which can be rather frustrating at times. So to George Nikolaidis and David Anear, thank you.

Garnett Znidaric

****

FOR SALE

SUPERBOARD II in an Aluminium Case with mains filtered 5 amp supply, 8K RAM, DABUG III, 48 character screen mod, 10 cassettes of software.
All manuals and circuits. * $300.00 *

Ring Roger Godfrey,

My name is Karl Valentic and I am the vice-president of M.A.C.E. I will be running M.A.C.E.V. while Gerry is on holidays.

At the moment I am writing a Disk Magazine for the ATARI computer. If you have any suggestions for the Disk Magazine, please let me know and I shall try to include them in the first issue.

Below is a review on a new book on the Assembler Editor. Each month I hope to bring you reviews on software, hardware and books for the ATARI.

PRODUCT REVIEW

There are a great number of books being written on the ATARI computers. Here is a review on one of them.

"THE ATARI ASSEMBLER" by DON AND KURT INMAN

This book contains 270 pages on how to program in assembly language, and use the ATARI Assembler cartridge.

The book is written in a simple language so anyone can understand it. At the end of each chapter, there are exercises and questions for you to complete. You do get the answers as well. You will learn how to use machine language routines from BASIC and do hexadecimal to decimal conversions. Using machine language routines from BASIC will improve your graphics animation and give you a higher quality in sound routines.

PRICE $13.50 from McGills Newsagency. Rating ***

RATINGS	* Fair
	** Good
	*** Very Good
	**** Excellent

For the next 3 months please refer all questions to:-

Karl Valentic

A TUNE FOR THE ATARI:

Enter the following program and see if you recognize this well known tune.

```
 10 DIM A(40)
 20 I=1
 30 READ A: IF A<>255 THEN A(I)=A: I=I+1: GOTO 30
 40 FOR J= 1 TO I-1
 50 SOUND 0,A(J),10,10
 60 FOR W= 1 TO 100: NEXT W
 70 NEXT J
 80 END
 90 DATA 121,121,108,96,121,96,108,162,121,121
100 DATA 108,96,121,121,128,162,121,121,108,96
110 DATA 91,96,108,121,128,162,144,128,121,121
120 DATA 121,121,255
```

K. E.

FOR SALE

ATARI 822 THERMAL PRINTER $300.00 contact Karl Valentic

ATARI'S REAL TIME CLOCK:

Most micro computers available now have some form of real time clock. The Atari computers are no exception. In the Atari computers there are three bytes which are strung together as a twenty four bit counter. This counter is incremented every fiftieth of a second by the Operating System. The three bytes in question are bytes 18, 19 and 20 (decimal).

If you would like to see a program which runs a real time clock using these bytes then you need look no further than your KAOS library. Jannene has two copies of Kilobaud (4/81) which contains this and other information on the Atari.

K. E.

****

EXTENDED MONITOR SEARCH ROUTINES N AND W

When using the search routine to find data etc. that's in many places, the data and address has to be entered for each search. I have altered this so that it shows the next occurrence by pressing 'N'. To fit it in I have altered the 'Z' routine which now needs (but does not use) a 'FROM' and 'TO' address.

THE CHANGES ARE:-

AT	OLD	NEW	AT	OLD	NEW
0D71	A5	4C	0FB8	11	1C
0D72	DC	BA	0FBE	DB	DD
0D73	85	0F	0FC3	DA	DC
0D74	DA	C9	0FE3	DA	DC
0D75	A5	4E	0FEB	DA	DC
0D76	DD	F0	0FEF	DB	DD
0D77	85	E2	0FFE	1F	74
0D78	DB	4C	0FFF	08	0D
0D79	A0	1F			
0D7A	00	08			
0D7B	4C	FF			
0D7C	9F	FF			
0D7D	0B	FF			

John Whitehead

****

KAOS LIBRARY NEWS:

KAOS library now has quite a large range of magazines for members to borrow. Currently I am composing a list of magazines held by the library which will be issued to members. The list will be updated as new magazines become available.

The library is gaining good response from those KAOS members who can read. If you are searching for a particular magazine then ring me (the number is on the front of the newsletter) and ask me if the library has access to it. Chances are that it has - although no promises are being made. If you have read an article which you feel may be of interest to other KAOS members then please let me know, so that I can let them know.

Although I have said that the library is large, it is still not too large that all magazine donations will not be gratefully received.

Jannene

THE EPROM PROGRAMMER

This programmer is one I designed 3 years ago with a few modifications to allow programming 2732's etc..

The programmer has proved very successful as most of the DABUG proms and other KAOS proms are burnt on this machine.

The burner requires a PIA or other latched port to interface it to your computer. It is only intended to burn the single supply EPROMS now available ie. 2708, 2716, 2532 (TMS), 2732 and 2764.

It is also versatile enough to read most ROMS ie. BASIC from the OSI, CHARGENS, MONITERS etc. provided that they are the 24 pin variety, which most are.

The only disadvantage with this type of programmer is that you have to program a 1K, 2K or 4K block in sequence ie. from first address to last address.

HOW IT WORKS

A personality header is put in the 14 pin socket which is wired to suit the particular prom you wish to read/program.

The row of 74193 chips supply the address you wish to see and the data is supplied by the A side of the PIA. ie.

Suppose you wish to read a 2716 EPROM. After plugging in the correct header and the eprom, the A side of the PIA is made to read data (all inputs)

The address counters are cleared to zero by applying a ZERO TO PBI and a HIGH TO ALL OTHER PB lines on the B side of the PIA. As the address is now ZERO you can read the data at this address from side A and put it in memory.

Next you remove the low from PBI and take PB0 to low causing the counters to increment to 001. Read port A and place in memory.

Keep pushing PB0 and reading result until complete contents of ROM are in memory. The programmer will tell you when you have exceeded the size of the ROM by taking the PB4 line high. (EOM)

To program an EPROM you make the A side of the PIA all O/Ps, clear the address registers and put data out on the PIA A side. You then take PB3 line low switching on the 25V for programming. (This should be low for all addresses. Don't pulse in up and down untill programming is complete.)

To program the data into the data location, take PB2 low for 50 msec and then take it back to high.

Increment the address counter and put new data out on side A then take PB2 low again for 50 msec. Do this till all data has been written into the EPROM.

Finally take PB3 high, removing the 25 volts. You can now read the EPROM and compare it to your original code.

If you did not understand the above ravings, don't worry to much, as there is a software driver available from the club. The software driver is very simple to operate and has a menu ie.

CC, PROG, EC, ED, SP, X

On entering the program you will be asked to supply the address of the START and END of the memory of the code you wish to program into the rom or where you wish to read rom contents into,ie. if you are programming a 2K EPROM you should supply 2K memory area ie.

BAM = 2000 (BAM = beginning of memory)
EAM = 27FF (EAM = end of memory)

ALL IN HEX PLEASE

FOR 2732 :-

BAM = 2000
EAM = 2FFF = 4K

THE MENU WILL APPEAR :-

CC	CHECK PROM IS CLEAR (yes if all locations are at FF hex)
PROG	PROGRAM TAKE CONTENTS OF MEMORY STARTING AT ABM AND PUT INTO EPROM UNTILL EAM REACHED
EC	ERROR CHECK- CHECK CONTENTS OF EPROM AND COMPARE IT TO MEMORY STARTING AT BAM. (This is automatically done after PROG or SP) COMES BACK WITH COUNT OF ERRORS.
ED	ERROR DUMP- DUMP TO SCREEN ERRORS BETWEEN BAM - EAM WITH CONTENTS OF EPROM.
SP	STORE PROM- STORE CONTENTS OF EPROM INTO MEMORY STARTING AT BAM AND STOPPING AT EAM OR UNTIL EPROM RUNS OUT OF LOCATIONS. THEN DO ERROR CHECK. USED TO READ ROM INTO MEMORY TO MAKE A COPY OR MOD.
X	EXIT PROGRAM

If you have a need for a programmer then the boards for this are available from either GEOFF COHEN (OMEGA) 72 SPOFFORTH ST., HOLT, A.C.T. 2615 or DAVID TASKER (TUGO) 111 BASS HIGHWAY, WESTBURY, TASMANIA 7303. They also carry software to drive the beast.

Both boards have an on board 25V supply derived from the 5V supply rail for programming, however you can also supply the 25Volts at 200ma from a separate supply (which is what I prefer to do). The unit gets its +5V from the computer.

David J. Anear

****

YET ANOTHER FLOPPY POWER SUPPLY

Power supplies are rather cumbersome objects which are a very important part of the system. While waiting for my floppy controller to be aligned, I put a bit of thought into where I was going to put my power supply. Simple, just add it to the Tasker Buss system (sort of).

Using a piece of Laminex backing board, I made up a small board the same shape as a RAM card so that once made up it could be slipped into it's slot on the Motherboard. I supplied and removed power from the board through a 12 pin Molex plug and socket.

The circuit that I used was based on the diagram shown in KAOS newsletter 1:2:4. The only difference being that I doubled up (or tripled in the case of 12VDC) on the voltage regulators so that they could handle all the current themselves. As a result these regulators share the load and also possess the ability to cope with whatever expansion I care to add to the Tasker Buss system.

For a heatsink on the 3 X 7812 regulators I used a piece of Aluminium door track. It was the right shape, and it works.

Garnett Znidaric

SUPERMON - COMMAND HANDLING

This month we will see how SUPERMON accepts and exeeutes a command.

Assume that the SYM has just been reset. Unless the reset routine has been modified, the SYM will preset the system RAM as described in the second article and will wait for an input from the user. This will allow the monitor to decide whether to communicate with the hex keypad and seven segment displays or a terminal. Pressing 'Q' on the terminal not only indicates the device type, but also allows the SYM to measure the baud rate and adjust itself accordingly. Obviously, pressing a key on the hex keypad instructs the SYM to use these keys and the seven segment displays. Since the seven secment displays require constant scanning, this routine is included with the keyboard input routine.

The SYM now jumps into its main loop. Here it JSRs to GETCOM, where it prints a '.' and waits for an input from the keyboard. It does this via the usual INCHR routine which vectors off to the appropriate input routine via INVEC in the system RAM.

The monitor now checks to see if the character entered is an 'S', 'L', or a 'U'. These characters begin the two character commands 'S1', 'S2', 'SP', 'L1', 'L2', 'LP', and 'U0' to 'U7'. In order to simplify the command analysis, these commands are 'hashed'. This means that two ASCII codes are combined to form a single byte code with the use of a hashing routine.

The SYM now has a single byte code representing the command entered. The system now outputs a space and jumps to subroutine PARM to fetch the parameters. This routine starts by clearing the parameter stack and setting the number of parameters entered (PARNR in system RAM) to zero. When a key is pressed. the character is checked to ensure that it is a hex number or a valid delimiter. If the key is in error, subroutine PARM is terminated with the ASCII code for the character in the accumulator.

If the key is a hex number, the monitor converts the ASCII code to a 4 bit binary number and shifts it into the right hand side of P3 in the system RAM. Since the parameter stack is only 16 bits wide, it will hold only the last four digits entered. The rest are lost.

If the key was a ',' or a '-', the stack is pushed using the subroutine PSHOVE which loses P1, moves P2 to P1, P3 to P2 and zero to P3. On returning to PARM the parameter counter PARNR is incremented. If PARNR exceeds 3 subroutine PARM is aborted with the ',' or '-' in the accumulator.

Subroutine GETCOM is now complete and control returns to the main loop where subroutine DISPAT is called. This routine first compares the accumulator contents with $0D (carriage return). As you may have noticed, there is no way of ending PARM without inducing an error. Since the accumulator will always contain the erroneous character, DISPAT must check for the carriage return terminating the command. If the accumulator contains anything else the error printing roútine is clled via URVEC (SUPERMON V1.0) or URSVEC (SUPERMON V1.1) in the system RAM. This routine prints 'ER' followed by the contents of the accumulator. The other function of DISPAT is to dispatch the command to the execution routines. The execution routines are split into four groups; one group for commands with no parameters, a group for commands with one parameter, etc. This means that the Memory examine and modify command appears in all four groups.

If after searching all of the commands for a given number of parameters no match is found, the error routine is called via URCVEC in the system RAM.

From here on the program execution depends on the command entered. Therefore, there is no need to go into the software any further.

Next Month:- SYSTEM RAM CONTENTS AND THE CASSETTE I/O SUBROUTINES

Between $A600 and $A67F is a block of RAM which is the key to SUPERMON's flexibility. This RAM contains pointers and data which can be used to customize SUPERMON and therefore, an understanding of its contents is necessary. Also we will be looking at the cassette I/O routines mentioned last month.

Brian Campbell

****

GLASS TYPWRITER PROGRAM

The short program listed below, is quite simple, and is for those people who don't have a word processor, but would still like to use their printer and computer as a typewriter.

The program has destructive back space and enables you to get a line the way you want it before it is sent out to the printer. It can also be easily modified to save what is written on to disk.

It is primarily written for disk systems, but there must be a similar keyboard routine in ROM for cassette based systems. Also the DISK!"GO 252B" command can be used in the same way as the INKEY command of some BASICs. Allowing non printing input for secret pass words, user interactive games (no CR.LF after-user response is required), etc.

```
  1  DISK!"CL"
  5  LI$=""
 10  DISK!"GO 252B"
 20  CH=PEEK(9815)
 30  IF CH=13 THEN PRINT:GOTO 200
 35  IF CH=127 THEN 100
 40  CH$=CHR$(CH)
 50  LI$=LI$+CH$
 55  IF LEN(LI$)>=63 THEN LI$=LEFT$(LI$,63)
 60  PRINT CHR$(13);LI$;
 99  GOTO 10
100  NL$=LEFT$(LI$,LEN(LI$)-1)
110  LI$=NL$: GOTO 60
200  PRINT #1,LI$
210  GOTO 5
```

Steve Stokes

****

FOR SALE

OSI SUPERBOARD Series 2 in Metal Case, 24K RAM, I/O Board, DABUG III monitor, includes 3 pot joystick interfaces (not switch type). Switchable 300/600 baud rates, 2 MHz version. All manuals and software. $500.00

Ring Paul Coburn

FIXES FOR THE FORTH MINI FLOPPY SYSTEM

People who have attempted to convert the Technical Products OSI-FORTH to run on 5¼" disks have found that it does not LOAD screens reliably. The problem appears to be caused by the fact that OSI's 5¼" disks cannot reliably store 8 sectors on a track, the last sector can overwrite the first sectors track header.

My solution is to change FORTH's use of 4 256 byte sectors per screen to one 1024 byte sector per screen with two sectors per track. This arrangement works reliably and is also much faster than the existing method.

The procedure is as follows:-

1. Stop FORTH accessing the disk for error messages.
2. Change - the byte per sector constant (B/BUF)
   - the buffers per screen variable (B/SCR)
   - the disk block offset variable (OFFSET)
   - the +BUF Word for new block size
3. Change DR0 to set OFFSET to the correct value
4. Change R/W (read/write word) to one 1024 byte sector block and correct the disk block limits
5. Set disk buffer locations to the new size using a modified CHANGE word.
6. Initialize the editor, type in the error messages on screens 3 and 4.

COMMANDS

```
1.      0     WARNING    !

2.      1024     '     B/BUF     !

        1        '     B/SCR     !

        2A             OFFSET    !

        4              13DA      C!

3.      2A             143C      C!

4.      4E             172A      !

        4C             1734      !

        2              175A      C!

        4              1778      C!

For one single-sided drive - 4E   171C   !
For one double-sided drive - 9C   171C   !

                             3    1739   C!

For two single-sided drives - 9C   171C   !
```

5. Enter top of memory (8000 for 32K, 6000 for 24K and 4000 for 16K systems)
   Enter number of disk buffers required. Two is suggested for 16K systems, 2 - 5 for 24 - 32K systems.

   You should have entered two numbers as appropriate,
   (For example - 6000 4)

ENTER

```
 R 80  -  DUP  DUP  r   B/BUF  4  +  *
```

This instruction should be entered on <u>one</u> line.

```
ENTER
-    DUP   DUP
'    FIRST   !   USE   !   PREV   !
'    LIMIT   !   210   !
DISK   "SA   16,1 = 0200/8"
DISK   "SA   18,1 = 1200/8"
COLD    0    WARNING    !
6. HEX : BLOCK-INIT CLR
   ." INITIALIZING BLOCKS - " CR
   FIRST 400 BLANKS SWAP
   DO FIRST I OFFSET @ + CR
   0 R/W LOOP DRO . "END" CR ;
   : SYS-INIT DRO 0 36 BLOCK-INIT ;
   DECIMAL      SYS-INIT      COLD
```

7. Using editor, enter error messages on screens 3 and 4. Then enter COLD .
The system should be up and running!
The BLOCK-INIT and SYS-INIT words above should be typed on to a screen and used to initialize all other FORTH disks.

David Wilson

FORTH ON 32 AND 48 CHARACTER PER LINE SYSTEMS

To modify FORTH from the standard 64 characters per line you need to change the FORTH characters per line constant C/L.

```
For 32 chars, enter - 32  '  C/L  !
For 48 chars, enter - 48  '  C/L  !
```

With these line lengths you will want to increase the number of lines in an EDIT screen to make better use of the 1024 byte screen size.

With 32 characters you could have 32 lines (32X32= 1024) but you will probably want to limit the size to the number of lines you can clearly see on your video. LET N = the number of lines you want.
ENTER

```
N  HEX  1A4C  C!  DECIMAL
```

Then do a system copy to save the modified FORTH.

With 48 characters you can have 21 lines on EDIT SCREEN (21X48= 1008)
ENTER

```
21  HEX  1A4C  C!  DECIMAL
```

You will lose the last 16 bytes of any previously set up screens.

To change the line line limit test in LINE
For 48 characters:-

PTO

```
ENTER

HEX 14 1B58 !

A3F 1B5A !
```

For 32 characters:-

```
ENTER

HEX A3F 1B5A !

DECIMAL N HEX 1B58 !
```

(N=number of lines you want)

These changes will make any previously set up screens look messy as lines will no longer be aligned, but these screens will still load.

David Wilson

****

COMMAND FILES AND DISK BASIC

A command file is a file of instructions which is automatically executed by the computer as a substitute for instructions typed in one at a time on the terminal. On the Challenger-1P/Superboard, this effect can be achieved by changing the input device from the terminal to either a disk file or a file in memory which has been set up (typically by a BASIC program) with the appropriate commands. The last thing the command file should do is to return control to the terminal (that is, change the input device back to the terminal). To illustrate this technique, I have written a program which should make the conversion of programs from 'standard' DOS (version 3.2 or 3.0) to George's 'extended' DOS into a painless operation. The target program is LOaded into memory and then LISTed to a disk file. You then boot up George's DOS, set aside room for one disk buffer, open the text file, and change the input flag to the disk file. After the program loads in the new DOS, it may be SAved in the normal manner.

The program has been written for a C1P with mini-floppies, and 32K of RAM and so some numbers may need to be changed for other systems.

```
   5  REM set memory I/O to the command file
   6  DISK! "ME 6000,6000"
  10  REM set up useful string constants
  12  Q$=CHR$(34)          quote character
  14  D$="DISK!"+Q$     DISK!"
  16  A$=CHR$(13)          carriage return
  18  P1$="?#6,"           print to DISK file
  20  P2$=P1$+Q$
  22  P$=P2$+D$+";CHR$(34);"+Q$
 100  INPUT"File name";N$
 110  B$=D$+"LO"+N$:GOSUB 2000
 200  INPUT"Text file name";N$
 210  B$="DISK0,6,"+Q$+N$:GOSUB 2000
 300  B$="LIST#6";GOSUB 2000
 400  B$=P2$+"EOF***":GOSUB 2000
 500  B$="DISKC,6":GOSUB 2000
 600  POKE8999,112:POKE9001,120     relocate disk file buffer
 700  B$="POKE8999,50:POKE9001,58":GOSUB 2000
 800  B$=D$+"IO,02,02":GOSUB 2000
1000  DISK!"IO 10"      start executing command file
1900  END  stop executing BASIC and start command file.
1990  REM  add a line to command file.  Lines are terminated by CR
1991  REM  and not CR/LF to satify DOS and Assembler
1992  REM BASIC ignores line-feeds, but better safe than sorry.
1993  REM #
2000  PRINT5,B$;A$;
2010  RETURN
```

The command file that this program generates is as follows:

```
DISK!"LO  basic-file-name
DISK 0,6," text-file-name  (note that the disk buffer has been
           relocated to 707E to 787E)
LIST 6
? 6,"EOF***
DISK C,6
POKE 8999,50:POKE9001,58   (restores disk buffer location to its
                           normal position of 327E to 3A7E)
DISK!"IO 02.02             return control to the keyboard
```

After you boot-up your new DOS, the commands to retrieve the program are:

```
POKE8999,112:POKE9001,120     move the disk buffer to a safe place
NEW
DISK 0,6," text-file-name
DISK!"IO 20
```

Control returns to the keyboard when the EOF*** is read. Lastly, save the program, and restore the buffer position:

```
DISK!"SA new-basic-file-name
POKE 8999,50 : POKE 9001,58
```

Further variations to this theme can be written to do such things as altering the number of disk-buffers preceding the program to a number nominated by you (0,1 or 2) and probably others I haven't thought of yet. Happy experimenting.

Next month: Corrections, modifications and improvements to DISK BASIC.

Rodney Eisfelder

****

OOOPS!

Owing to the haste with which I wrote the article on the C4 conversion board, there were a few omissions and corrections. First the corrections:-

Move the jumper that is on pin 4 of the 16 pin socket from pin 6 to pin 8 of U41 on the Superboard.
Move the jumper that is on pin 11 of the 16 pin socket from pin 12 to pin of 2A on the C4 board. Disconnect the 1K resistor from pin 3 of the 16 pin socket and solder to pin 2.
Move the jumper from pin 2 of the 16 pin socket to pin 3 of the same.
Omissions:-
On the Superboard connect pin 14 of U60 to pin 2 of 16 pin socket
Connect pin 13 of U60 to pin 3 of 16 pin socket
Connect pin 12 of U60 to pin 6 of 16 pin socket
Now on the C4 board connect ground to pin 14 of the 14 pin socket.
With a little bit of luck (!!!!!) the new format should work but to make the new format usable, you still have to make a few modifications to the software and hardware.

To make the machine look like a C1 you need to invert the keyboard,and burn a 4K Eprom with the C1 Dabug in the lower 2K and a C4 Dabug in high memory. However, to describe the hardware necessary will require diagrams and a detailed explanation and as space does not permit it in the current issue, it will be held over for a later issue.

PTO

To use the new format with the C4 keyboard, you copy the C1 and C4 monitors into memory, then copy the C4 keyboard routine down over the original C1 routine. The keyboard routine sits in the same position on both monitors. Burn this into a 4K Eprom, remove the original monitor and replace with the new one. From the underside of the Superboard, remove the jumper from between pin 21 and +5 volts of monitor socket. Connect pin 21 of the monitor to pin 1 of the 14 pin socket.

Now when you select the C1 screen format, it will also swap which 2K monitor is being used. After you change format you may have to do a warm start to recover from the swap as the CPU can get lost.

NOTE The end of the 5th line on page 13 should read ' from pin 12 to pin 14 '

Jeff Rae

****

To get back to Warm Start BASIC after using the Extended Monitor:

Press	Break		
"	C (Don't hit RETURN)		
"	Break		
"	M		
Change	0000	to	4C
	1		74
	2		A2
	3		4C
	4		C3
	5		A8

This may also work for BASIC crashes that can't be fixed in the usual way.

John Whitehead

****

MACHINE CODE PROGRAMMING

This month we will be describing a method of hooking the machine code routines that you are writing ??? into BASIC.

There is a routine at $00BC which is used by BASIC to get the next character from the current line, by putting a jump address in $CE $CF you can divert control to your routine.

The first seven lines of the program set up this address and return to warm start. The next two lines look for the # and diverts control to the HOOK routine, if it is found. If the test fails, the next five lines duplicate the code in the $00BC routine and returns to BASIC for processing of the character in the Accumulator.

The HOOK routine increments the Y counter and gets the next character from the BASIC line, it then looks for a match and branches to the appropriate address eg. 43 hex (C) Clearscreen. If a match is not found the routine at $AC0C prints the SN error message.

The HOOK command can be used in the immediate or program mode, in the program mode, if it is the first statement on the line, it must be proceeded by a colon eg. 10 :#C:PRINT"ABC"

The remainder of the program consists of two routines to demonstrate how the hook works. The first is a screen clear that loads the accumulator with $20 (space) and stores it in the screen RAM $Dxxx,Y increments Y and loops until Y equals zero. The second routine gets characters from the cassette port and displays them on the screen with out putting them in memory. Hitting the space bar returns control to BASIC.

NOTE If you have a Dabug you can use its Screen Clear routine. Replace the code from $0254 to $0276 with - 0254 20 D5 FC JSR $FCD5

```
0222  A94C              LDA #$4C             Load JMP instruction and
0224  85CD              STA $CD              address into $CD $CE & $CF
0226  A931              LDA #$31
0228  85CE              STA $CE
022A  A902              LDA #$02
022C  85CF              STA $CF
022E  4C0000            JMP $0000            Jump to Warm Start
                        ;
0231  C923              CMP #$~~43~~ 23      Is it a #
0233  F007              BEQ HOOK             Yes, go to Hook routine
0235  38                SEC                  No, do normal routine and
0236  E930              SBC #$30             return for next character
0238  38                SEC
0239  E9D0              SBS #$D0
023B  60                RTS
                        ;
023C  A001     HOOK     LDY #$01
023E  B1C3              LDA ($C3),Y          Get character following
0240  C943              CMP #$43             Is it 'C'
0242  F010              BEQ CLRSCR           Yes, go to Screen Clear
0244  C956              CMP #$56             Is it 'V'
0246  F032              BEQ VIEW             Yes, go to View
0248  4C0CAC            JMP $AC0C            Illegal character, go to
                                             error message routine
                        ;
024B  E6C3     ADDONE   INC $C3              Add one to character/line
024D  D002              BNE RETURN           count and return for next
024F  E6C4              INC $C4              character
0251  4CBC00   RETURN   JMP $BC
                        ;
0254  98       CLRSCR   TYA                  Transfer Y to A and
0255  48                PHA                  push A onto the Stack
0256  A920              LDA #$20
0258  A000              LDY #$00
025A  9900D7   LOOP     STA $D700,Y
025D  9900D6            STA $D600,Y
0260  9900D5            STA $D500,Y
0263  9900D4            STA $D400,y
0266  9900D3            STA $D300,Y
0269  9900D2            STA $D200,Y
026C  9900D1            STA $D100,Y
026F  9900D0            STA $D000,Y
0272  C8                INY
0273  D0E5              BNE LOOP
0275  68                PLA                  Pull A from stack and
0276  A8                TAY                  transfer it to Y
0277  4C4B02            JMP ADDONE
                        ;
027A  A9FD     VIEW     LDA #$FD *02
027C  8D00DF            STA $DF00
027F  A910              LDA #$10
0281  2C00DF            BIT $DF00
0284  F0C5  *D0         BEQ ADDONE   *BNE
0286  AD00F0 *FC        LDA $F000    *FC00
0289  4A                LSR A
028A  90EE              BCC VIEW
028C  AD01F0 *FC        LDA $F001    *FC01
028F  202DBF            JSR $BF2D
0292  4C7A02            JMP VIEW
                                 * = Values for C4
```

THE MEETING WAS KAOS

VIDEO BOARD: Those who have memories would recall the long promised all dancing, all singing, video board. Well, HALLELUJAH, it's being produced at present. To jolt those memories, the video board will support the original C1P screen format, C4P 64X32, C4P 32X32 formats with guard bands as well as switchable keyboard.

OSI EXPANSION: The meeting revealed that Bill Chilcott's new expansion board (covered in last month's newsletter) may take a slightly new direction, with talk of making the board support 64K of CMOS RAM. Hopefully, we will learn of any new developements before the next newsletter.

COMMADORE VIC: For $399.00 it could be seen as a competative alternative to the Superboard. Without becoming too bogged down with comparisons between computers, I must point out that the VIC has a number of features such as colour and Hi-Res graphics which becomes quite attractive for those who expect value for money. The only criticism I have at present is the requirement of a special digital recorder which costs around $100.00, though I'm told that a few of the members have nearly completed a simple interface which will connect to any recorder.

DISK SOFTWARE: Jeff Rae gave a talk on problems associated with 5¼" and 8" disks, these included Pascal, Forth and down loading programs from T.A.B.

If the interstate and country members could imagine ten or so computers scattered around two large rooms on the second storey of a primary school, with approximately a hundred people, hussling, bustling, exchanging ideas, demonstrating their programs and modifications, they could certainly imagine the atmosphere generated at the MARCH meeting.

Hope to see you in April,
73's Rod Drysdale
VK3BYU

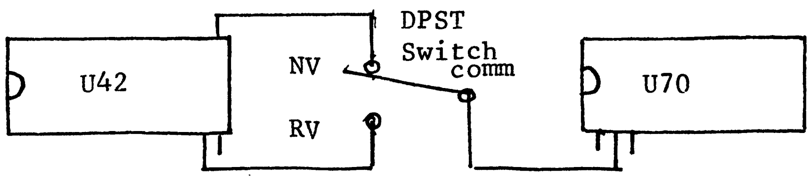
****

REVERSE VIDEO (Manually switched type)

Will suit Superboard 2 (600 board) or any other board which uses a 74LS165 shift register to serialize the output of the character generator.

PARTS REQUIRED:- One SPDT switch to suit panel decor, 3 lengths of wire. Also the usual computer surgeon's tools for delicate brain surgery.

MODUS OPERANDI:- (described for 600 board, others please improvise):(PLEASE READ THROUGH TWICE, THEN PROCEED)
(1) Find the 74LS165 (U42), next to character generator
(2) Cut track running from U42 pin 9 to U70 (7403) pin 2 at a conveient point
(3) Using fine insulated wire correct as follows:-
   a. Switch common to U70 pin 2
   b. One switch pole to U42 pin 9 (Q): normal video
   c. Other switch pole to U42 pin 7 ($\overline{Q}$): reverse video



NOTE: Your monitor <u>may</u> need some adjustment to brightness and contrast to optimise the screen.

Ralph Hess

****

Line 55 of the Glass Typewriter program on page 9 should read:
```
55 IF LEN(LI$)>=63 THEN LI$=LEFT$(LI$,63)
```